

# UM APLICATIVO CONTADOR DE MOEDAS UTILIZANDO TÉCNICAS DE PROCESSAMENTO DIGITAL DE IMAGENS

Denis Ricardo da Silva Medeiros  
Pedro Henrique de Medeiros

# Sumário

- Introdução
- Visão geral do projeto
  - Proposta, restrições e limitações
- Estratégia utilizada
  - Segmentação e rede neural artificial
- Resultados
- Considerações finais

# Introdução

- Formas de pagamento eletrônico vs dinheiro tradicional
- Contagem do dinheiro tradicional, dependendo do montante, pode ser árduo e demorado, principalmente com moedas





# Introdução

- Atualmente, câmeras são amplamente acessíveis e estão presentes em celulares, computadores e outros dispositivos
- Por que não utilizar esses dispositivos para resolver o problema da contagem de dinheiro?



# Visão geral do projeto

- Proposta do projeto
  - Desenvolver uma aplicação capaz de contar quantos reais há em uma imagem com várias moedas utilizando técnicas de processamento digital de imagens e de inteligência artificial (rede neural artificial como classificador)
  - Moedas possuem como características úteis cor e tamanho, mas podem facilmente estar rotacionadas.

# Visão geral do projeto

- Proposta do projeto
  - Aplicação dividida em 4 módulos:
    - 1. Segmentador (prepara as moedas para o conjunto treinamento)
    - 2. Calibrador (treinamento da rede neural artificial)
    - 3. Validador (ajuda a validar a efetividade da rede neural)
    - 4. Contador (programa que, utilizando o conhecimento aprendido pela rede neural artificial, consegue contar quantos reais há em uma imagem com várias moedas)

# Visão geral do projeto

- Restrições e limitações
  - As imagens com as moedas devem ser adquiridas com resolução, cor de fundo e iluminação adequada e elas não podem estar sobrepostas
    - Cor de fundo: branco
    - Distância da câmera até às moedas: 30 cm ou o suficiente para enquadrar uma folha de papel A4 na imagem
  - Serão usadas apenas 3 tipos de moedas:
    - 25 centavos, 50 centavos e de 1 real



# Visão geral do projeto

- Exemplo de imagem atendendo às restrições:





# Estratégia utilizada

- Calibração do sistema
  - Obtenção de imagens modelos
  - Segmentação da imagem e busca por moedas
  - Separação de moedas já identificadas no passo anterior para o conjunto de treinamento da rede neural artificial e para o conjunto de validação
  - Treinamento da rede neural artificial
  - Validação da rede neural artificial

# Estratégia utilizada

- Funcionamento normal do sistema
  - Obtenção de imagem
  - Segmentação da imagem e busca por moedas
  - Verificação do valor de cada uma através da rede neural artificial
  - Exibição do valor total contado pelo aplicativo

# Segmentação

- 1. Suavização da imagem através de filtro gaussiano
- 2. Operação de limiar utilizando o método de Otsu, para tentar encontrar todas as moedas
- 3. Remoção de moedas na borda da imagem
- 4. Fechamento morfológico (dilatação seguida de erosão) para conectar possíveis falhas na borda e fechar buracos, com elemento estruturante variando de 1 a 8.



# Segmentação

5. Remoção de eventuais resíduos dentro das moedas (só as bordas interessam)
- 6. Busca por contornos através de um algoritmo seguidor de borda
- 7. Criação de círculos (os menores possíveis) em volta dos contornos encontrados

—

# Segmentação

- 8. Verificação se o contorno encontrado realmente é um círculo
  - O raio do círculo deve ser maior que um limiar (50 píxels), para um ponto não ser confundido com um círculo
  - A relação entre área do contorno e a área do círculo que circunda esse contorno não pode ser menor que um limiar (70%).
- 9. São criadas imagens para cada moeda encontrada, cujas cores fora da borda são alteradas para branco.

# Segmentação

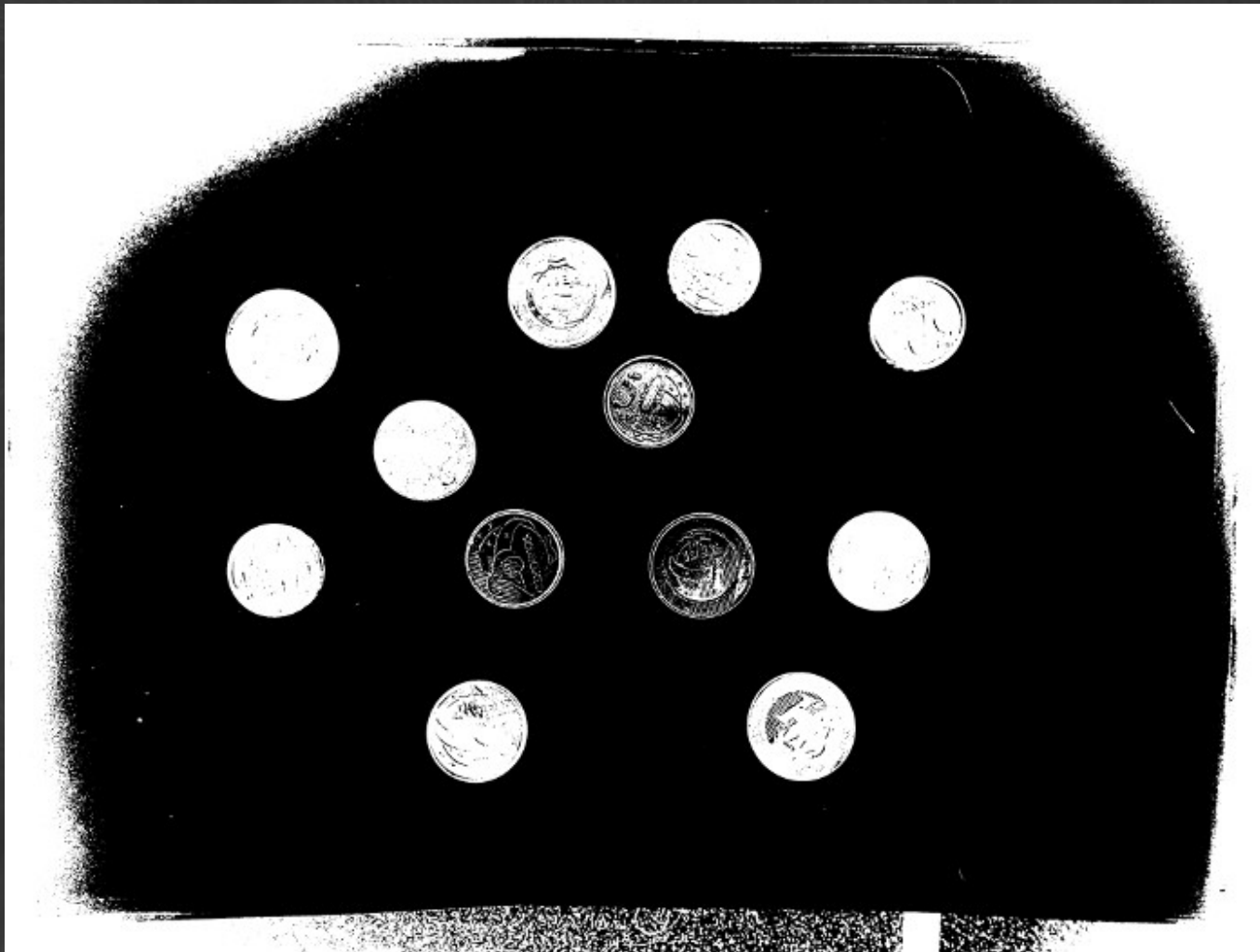
- Imagem obtida na câmera:





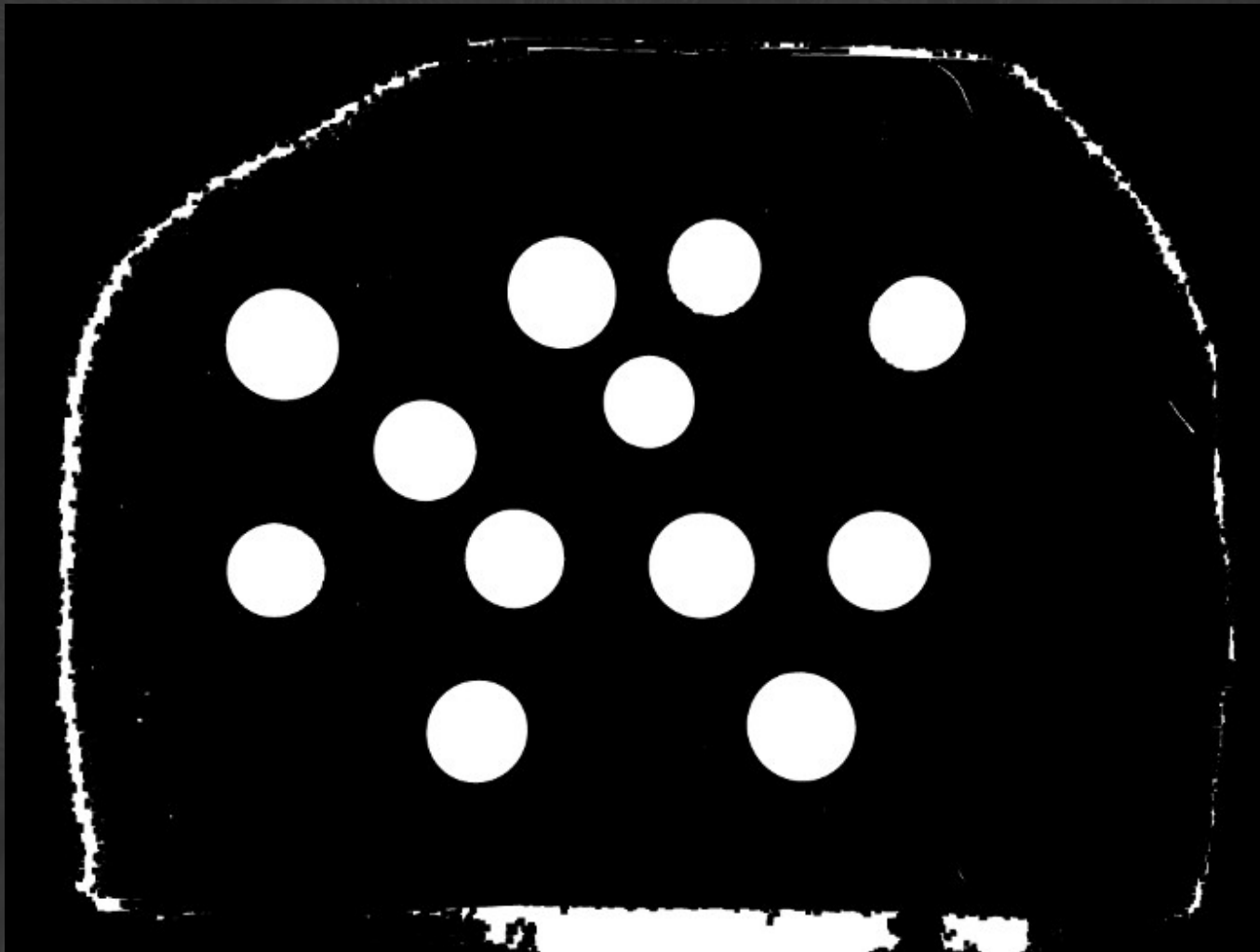
# Segmentação

- Imagem após a operação de limiar (Otsu):



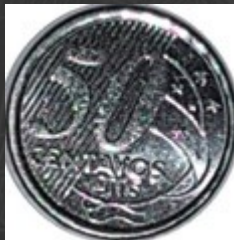
# Segmentação

- Imagem após a fechamento morfológico:



# Segmentação

- Imagens criadas após o processo:





# Rede Neural Artificial

- Rede neural artificial do tipo *Multilayer Perceptron* (MLP)
  - 4 camadas, sendo 2 escondidas com 32 neurônios cada
  - 41 entradas, sendo as 32 primeiras níveis do histograma da matiz da imagem já equalizada, as 2 seguintes a largura e altura da imagem e os 7 últimos os seus momentos invariantes
  - 3 saídas, cada uma para um tipo de meda, sendo 1 para verdadeiro e -1 para falso
  - Função de ativação: sigmoide simétrica (-1 a 1)
  - Número máximo de iterações: 10000
  - Erro máximo de parada:  $1e-8$
  - Taxa de aprendizado e momentum: 0.1

# Rede Neural Artificial

- Conjunto treinamento com 20 moedas de cada (25, 50 e 1 real), sendo 10 da face e as outras 10 do número.
- Tempo total do treinamento: 2.018s
- Conjunto validação: 4 moedas de cada tipo, sendo 2 da face e as outras 2 do número.
- Resultado da validação: 100% de acerto.

# Resultados

- Exemplo da validação (25 centavos):

```
denis@gerencia-notebook:/tmp/projetofinal$ ./validador validacao/25/1.jpg
-----
Moeda :
(25) = 1.00226
(50) = -0.997003
(100) = -1.0058
-----
denis@gerencia-notebook:/tmp/projetofinal$ ./validador validacao/25/2.jpg
-----
Moeda :
(25) = 0.992012
(50) = -0.997708
(100) = -1.00573
-----
denis@gerencia-notebook:/tmp/projetofinal$ ./validador validacao/25/3.jpg
-----
Moeda :
(25) = 0.498484
(50) = -1.13654
(100) = -0.386701
-----
denis@gerencia-notebook:/tmp/projetofinal$ ./validador validacao/25/4.jpg
-----
Moeda :
(25) = 1.00382
(50) = -1.00784
(100) = -0.990536
-----
```

# Resultados

- Exemplo da validação (50 centavos):

```
denis@gerencia-notebook:/tmp/projetofinal$ ./validador validacao/50/1.jpg
-----
Moeda :
(25) = -1.01978
(50) = 0.981279
(100) = -0.962827
-----

denis@gerencia-notebook:/tmp/projetofinal$ ./validador validacao/50/2.jpg
-----
Moeda :
(25) = -1.00145
(50) = 1.02011
(100) = -1.00438
-----

denis@gerencia-notebook:/tmp/projetofinal$ ./validador validacao/50/3.jpg
-----
Moeda :
(25) = -0.981349
(50) = 0.976257
(100) = -0.997547
-----

denis@gerencia-notebook:/tmp/projetofinal$ ./validador validacao/50/4.jpg
-----
Moeda :
(25) = -0.993451
(50) = 0.979165
(100) = -0.997587
-----
```



# Resultados

- Exemplo da validação (100 centavos):

```
denis@gerencia-notebook:/tmp/projetofinal$ ./validador validacao/100/1.jpg
-----
Moeda :
(25) = -0.980049
(50) = -0.932487
(100) = 0.899731
-----

denis@gerencia-notebook:/tmp/projetofinal$ ./validador validacao/100/2.jpg
-----
Moeda :
(25) = -1.02751
(50) = -0.966398
(100) = 0.984723
-----

denis@gerencia-notebook:/tmp/projetofinal$ ./validador validacao/100/3.jpg
-----
Moeda :
(25) = -1.07407
(50) = -0.97125
(100) = 1.02219
-----

denis@gerencia-notebook:/tmp/projetofinal$ ./validador validacao/100/4.jpg
-----
Moeda :
(25) = -1.01076
(50) = -0.992155
(100) = 1.00249
-----
```

# Resultados

- Exemplo de execução do contador
  - Imagem de entrada:



# Resultados

- Exemplo de execução do contador
  - Execução do contador:

```
denis@gerencia-notebook:/tmp/projetofinal$ time ./contador completas/6.jpg
[main] Realizando detecção pelo algoritmo padrão.
[detectar] Usando fechamento de tamanho 1
[detectar] Foram encontradas 12 moedas.
[detectar] Usando fechamento de tamanho 2
[detectar] Foram encontradas 12 moedas.
[detectar] Usando fechamento de tamanho 4
[detectar] Foram encontradas 12 moedas.
[detectar] Usando fechamento de tamanho 8
[detectar] Foram encontradas 12 moedas.
[main] O sistema detectou 7 reais nesta imagem.

real    0m2.299s
user    0m2.412s
sys     0m0.204s
denis@gerencia-notebook:/tmp/projetofinal$
```



# Considerações finais

- Importância na boa obtenção das imagens
- Aplicação de vários conceitos aprendidos no curso de Processamento Digital de Imagens
- Possibilidade de melhoria futura do projeto, passando a incluir todas as moedas do Real, com menos restrição na obtenção da imagem e possivelmente com o sistema funcionando em tempo real